Numer. Math. (2006) 105:299–313
DOI 10.1007/s00211-006-0042-1

Numerische
Mathematik

A note on the error analysis of classical Gram–Schmidt

Alicja Smoktunowicz · Jesse L. Barlow · Julien Langou

Received: 30 June 2006 / Revised: 30 September 2006 /
Published online: 3 November 2006

Abstract An error analysis result is given for classical Gram–Schmidt factorization of a full rank matrix A into $A = QR$ where Q is left orthogonal (has orthonormal columns) and R is upper triangular. The work presented here shows that the computed R satisfies $R^{\mathrm{T}} R = A^{\mathrm{T}} A + E$ where E is an appropriately small backward error, but only if the diagonals of R are computed in a manner similar to Cholesky factorization of the normal equations matrix. At the end of the article, implications for classical Gram–Schmidt with reorthogonalization are noted.

A similar result is stated in Giraud et al. (Numer Math 101(1):87–100, 2005). However, for that result to hold, the diagonals of R must be computed in the manner recommended in this work.

Jesse Barlow's research was supported by the National Science Foundation under grant no. CCF-0429481.

A. Smoktunowicz
Faculty of Mathematics and Information Science, Warsaw University of Technology,
Pl. Politechniki 1, Warsaw, 00-661 Poland
e-mail: smok@mini.pw.edu.pl

J. L. Barlow (✉)
Department of Computer Science and Engineering, The Pennsylvania State University,
University Park, PA 16802-6822, USA
e-mail: barlow@cse.psu.edu

J. Langou
Department of Mathematics,
University of Colorado at Denver and Health Sciences Center, Denver, USA
e-mail: langou@math.cudenver.edu

The classical Gram–Schmidt (CGS) orthogonal factorization is analyzed in a recent work of Giraud et al. [7] and in a number of other sources [3,10,14,1, 4,9,13, Sect. 6.9], [2, Sect. 2.4.5].

For a matrix $A \in \mathbb{R}^{m\times n}$ $(m \geq n)$ with $\mathrm{rank}(A) = n$, in exact arithmetic, the algorithm produces a factorization

$$A = QR \tag{1}$$

where Q is *left orthogonal* (i.e., $Q^{\mathrm{T}}Q = I_n$), and $R \in \mathbb{R}^{n\times n}$ is upper triangular and nonsingular. In describing the algorithms, we use the notational conventions,

$$A = (\mathbf{a}_1, \ldots, \mathbf{a}_n), \quad Q = (\mathbf{q}_1, \ldots, \mathbf{q}_n),$$
$$R = (r_{jk}).$$

The algorithm forms Q and R from A column by column as described in the following pseudo-code. We label this algorithm CGS–S, for classical Gram–Schmidt "standard."

Algorithm 1 (Classical Gram–Schmidt Orthogonal Factorization (Standard) (CGS–S))

$r_{11} = \|\mathbf{a}_1\|_2; \mathbf{q}_1 = \mathbf{a}_1/r_{11};$
$R_1 = (r_{11}); Q_1 = (\mathbf{q}_1);$
for $= k = 2 : n$
  $\mathbf{s}_k = Q_{k-1}^{\mathrm{T}}\mathbf{a}_k;$
  $\mathbf{v}_k = \mathbf{a}_k - Q_{k-1}\mathbf{s}_k;$
  $r_{kk} = \|\mathbf{v}_k\|_2;$
  $\mathbf{q}_k = \mathbf{v}_k/r_{kk};$
  $R_k = \begin{array}{c c} & \begin{array}{cc} k-1 & 1 \end{array} \\ \begin{array}{c} k-1 \\ 1 \end{array} & \begin{pmatrix} R_{k-1} & \mathbf{s}_k \\ 0 & r_{kk} \end{pmatrix} \end{array}; \quad Q_k = \begin{array}{c} \begin{array}{cc} k-1 & 1 \end{array} \\ \begin{pmatrix} Q_{k-1} & \mathbf{q}_k \end{pmatrix} \end{array};$
end;
$Q = Q_n; \ R = R_n;$

As is well known [2, p. 67, Sect. 2.4.5], in floating point arithmetic, Q is far from left orthogonal. As pointed out by Gander [5], as quoted by Björck [2, p.67, Sect. 2.4.5], "even computing Q via the Cholesky decomposition of $A^{\mathrm{T}}A$ seems superior to CGS." The authors of [7] prove a number of results about classical Gram–Schmidt. This note shows that for one of their results (Lemma 1 in [7]), the diagonal elements r_{kk} should be computed differently from Algorithm 1, substituting a Cholesky-like formula for r_{kk} rather than setting $r_{kk} = \|\mathbf{v}_k\|_2$. That change produces Algorithm 2. Since it uses a pythagorean identity to compute the diagonals of R, we call it CGS-P for "classical Gram–Schmidt pythagorean."

Algorithm 2 (Cholesky-like Classical Gram–Schmidt Orthogonal Factorization (CGS–P))

$r_{11} = \|\mathbf{a}_1\|_2; \mathbf{q}_1 = \mathbf{a}_1/r_{11};$
$R_1 = (r_{11}); Q_1 = (\mathbf{q}_1);$
for $= k = 2\colon n$
$\quad \mathbf{s}_k = Q_{k-1}^{\mathrm{T}} \mathbf{a}_k;$
$\quad \mathbf{v}_k = \mathbf{a}_k - Q_{k-1}\mathbf{s}_k;$
$\quad \psi_k = \|\mathbf{a}_k\|_2; \phi_k = \|\mathbf{s}_k\|_2;$
$\quad r_{kk} = (\psi_k - \phi_k)^{1/2}\,(\psi_k + \phi_k)^{1/2};$
$\quad \mathbf{q}_k = \mathbf{v}_k/r_{kk};$

$$R_k = \begin{array}{c} \\ k-1 \\ 1 \end{array} \begin{array}{c} \begin{array}{cc} k-1 & 1 \end{array} \\ \begin{pmatrix} R_{k-1} & \mathbf{s}_k \\ 0 & r_{kk} \end{pmatrix} \end{array}; \quad Q_k = \begin{array}{c} \begin{array}{cc} k-1 & 1 \end{array} \\ \begin{pmatrix} Q_{k-1} & \mathbf{q}_k \end{pmatrix} \end{array};$$

end;
$Q = Q_n; R = R_n;$

We assume that we are using a floating point arithmetic that satisfies the IEEE floating point standard. In IEEE arithmetic

$$f\ell(x+y) = (x+y)(1+\delta), \quad |\delta| \le \varepsilon_M$$

for results in the normalized range [12, p. 32].

Letting ε_M be the machine unit, we follow Golub and Van Loan [8, Sect. 2.4.6] and use the linear approximation

$$(1+\varepsilon_M)^{p(n)} = 1 + p(n)\varepsilon_M + O(\varepsilon_M^2)$$

for a modest function $p(n)$ thereby assuming that the $O(\varepsilon_M^2)$ makes no significant contribution.

For the sake of self containment, we give Lemma 1 from [7].

Lemma 1 [7] *In floating point arithmetic with machine unit ε_M, the computed upper triangular factor from Algorithm 1 satisfies*

$$R^{\mathrm{T}} R = A^{\mathrm{T}} A + E, \quad \|E\|_2 \le c(m,n)\|A\|_2^2 \varepsilon_M$$

where $c(m,n) = O(mn^2)$.

As stated, this lemma is not correct for Algorithm 1, but a slightly different version of this result holds for Algorithm 2. Since the proof of Theorem 1 in [7] depends upon Lemma 1, Theorem 1 cannot be established unless an alternative proof can be found.

We define the four functions

$$
\begin{aligned}
c_1(m,k) &= \begin{cases} 1 & k=1 \\ 2\sqrt{2}mk + 2\sqrt{k} & k=2,\ldots,n, \end{cases} \\
c_2(m,k) &= \begin{cases} m+2 & k=1 \\ 3.5mk^2 - 1.5mk + 16k & k=2,\ldots,n, \end{cases} \\
c_3(m,k) &= 0.5c_2(m,k), \quad c_4(m,k) = c_2(m,k) + 2c_1(m,k),
\end{aligned}
\tag{2}
$$

we let A_k be the first k columns of A, and let

$$\kappa_2(R_k) = \|R_k\|_2 \|R_k^{-1}\|_2.$$

The new version of Lemma 1 is Theorem 1.

Theorem 1 *Assume that in floating point arithmetic with machine unit ε_M, for the R resulting from Algorithm 2 for each k, we have*

$$c_4(m,k)\varepsilon_M \kappa_2(R_k)^2 < 1. \tag{3}$$

Let $A_k \in \mathbb{R}^{m\times k}$ consist of the first k columns of A. Then, for $k = 1,\ldots,n$, to within terms of $O(\varepsilon_M^2)$, the computed matrices R_k and Q_k satisfy

$$
\begin{aligned}
Q_kR_k - A_k &= \Delta A_k, \quad \|\Delta A_k\|_2 \le c_1(m,k)\|A_k\|_2\varepsilon_M, & (4)\\
R_k^{\mathrm{T}}R_k - A_k^{\mathrm{T}}A_k &= E_k, \quad \|E_k\|_2 \le c_2(m,k)\|A_k\|_2{}^2\varepsilon_M, & (5)\\
\|R_k\|_2 &= \|A_k\|_2(1+\mu_k), \quad |\mu_k| \le c_3(m,k)\varepsilon_M, & (6)\\
\|I - Q_k^{\mathrm{T}}Q_k\|_2 &\le c_4(m,k)\kappa_2(R_k)^2\varepsilon_M, & (7)\\
\|Q_k\|_2 &\le \sqrt{2}. & (8)
\end{aligned}
$$

The proof of Theorem 1 is given in the appendix.

The restriction (3) assures that R is nonsingular, and that (7) and (8) hold. A weaker assumption that assures that R is nonsingular and that $\|Q_k\|_2$ is bounded would yield bounds similar to (4), (5), and (6).

Remark 1 The condition (3) and the bound (7) are stated in terms of $\kappa_2(R_k)$. We now show how it may be stated in terms of

$$\kappa_2(A_k) = \|A_k\|_2\|A_k^{\dagger}\|_2$$

where $A_k^{\dagger}$ is the Moore–Penrose pseudoinverse of A_k. In exact arithmetic, $\kappa_2(A_k)$ and $\kappa_2(R_k)$ are the same quantity, and Eq. (6) states that $\|R_k\|_2$ and $\|A_k\|_2$ are nearly interchangable in floating point arithmetic. To relate $\|R_k^{-1}\|_2$ and $\|A_k^{\dagger}\|_2$, we use eigenvalue inequalities.

From the fact that

$$\|R_k^{-1}\|_2^{-1} = \sqrt{\lambda_k(R_k^{\mathrm{T}}R_k)}, \quad \|A_k^{\dagger}\|_2^{-1} = \sqrt{\lambda_k(A_k^{\mathrm{T}}A_k)} \tag{9}$$

where $\lambda_k(\cdot)$ denotes *kth* largest (and therefore smallest) eigenvalue, we can obtain an upper bound for $\|A_k^\dagger\|_2$ using Weyl's monotonicity theorem [13, Theorem 10.3.1]. Applying that theorem to (5), we have

$$\begin{aligned}\lambda_k(R_k^{\mathrm{T}} R_k) &\geq \lambda_k(A_k^{\mathrm{T}} A_k) - \|E_k\|_2 \\ &\geq \lambda_k(A_k^{\mathrm{T}} A_k) - \varepsilon_M c_2(m,k)\|A_k\|_2^2 + O(\varepsilon_M^2) \\ &= \lambda_k(A_k^{\mathrm{T}} A_k) - \varepsilon_M c_2(m,k)\|R_k\|_2^2 + O(\varepsilon_M^2) \\ &\geq \lambda_k(A_k^{\mathrm{T}} A_k)(1-\zeta_k)\end{aligned}$$

where

$$\zeta_k = \varepsilon_M c_2(m,k)\kappa_2(R_k)^2 + O(\varepsilon_M^2). \tag{10}$$

Using (9), we have

$$\|R_k^\dagger\|_2 \leq \|A_k^{-1}\|_2(1-\zeta_k)^{-1/2}.$$

From (6), we may conclude that

$$\kappa_2(R_k) \leq \kappa_2(A_k)(1+\mu_k)(1-\zeta_k)^{-1/2}.$$

Thus a slight variation of the condition (3) may be stated in terms of $\kappa_2(A_k)$. Since it fits more naturally into the proof of Theorem 1 and it is more easily computed than $\kappa_2(A_k)$, we use $\kappa_2(R_k)$.

The conclusion of Theorem 1 does not hold for Algorithm 1, as shown by the following example. We were able to construct several similar examples. Both examples were done in MATLAB version 7 on a Dell Precision 370 workstation running Linux.

Example 1 We produced a 6×5 matrix with the following MATLAB code.

```
B=hilb(6);
A1 = ones(6,3) + B(: ,1: 3) * 1e - 2;
B=pascal(6);
A2 = B(: ,1: 2);
A=[A1 A2];
```

The command hilb (6) produces the 6×6 Hilbert matrix, the command ones (6,3) produces a 6×3 matrix of ones, and the command pascal (6) produces a 6×6 matrix from Pascal's triangle. The condition number of R from Algorithm 2, $\kappa_2(R) = \|R\|_2\|R^{-1}\|_2$, computed by the MATLAB command **cond**, is $3.9874 \cdot 10^6$, thus given that $\varepsilon_M \approx 2.2206 \cdot 10^{-16}$ in IEEE double precision, R is neither well-conditioned nor near singular.

Table 1 Orthogonality and Normal Equations Error from CGS Algorithms for Example 1

Algorithm	$\|A^{\mathrm{T}}A - R^{\mathrm{T}}R\|_2/\|A\|_2^2$	$\|I - Q^{\mathrm{T}}Q\|_2$
CGS–S (Algorithm 1)	4.5460e-9	3.9874e-6
CGS–P (Algorithm 2)	3.3760e-17	5.2234e-5

We computed the Q–R factorization using Algorithm 1 (CGS–S) and then we computed the same factorization using Algorithm 2 (CGS–P). The resulting Q and R satisfy the results in Table 1. The work in this paper was motivated by noting that the components of $A^{\mathrm{T}}A - R^{\mathrm{T}}R/\|A\|_2^2 = O(\varepsilon_M)$ except for the $(4,4)$ and $(5,5)$ entries which were substantially larger.

The bound on $\|A^{\mathrm{T}}A - R^{\mathrm{T}}R\|_2$ in (5) appears to be satisfied if r_{kk} is computed as in Algorithm 2, but it is not if r_{kk} is computed as in Algorithm 1.

A larger, more complex, but better conditioned example is given next.

Example 2 A large class of examples where CGS-S obtains a large value of $\|A^{\mathrm{T}}\ A - R^{\mathrm{T}}\ R\|_2/(\|A\|_2^2)$, but CGS-P does not arises from glued matrices. A general MATLAB code for these glued matrices is given by

```
function [A]=create_gluedmatrix
 (condA_glob,condA,m,nglued,nbglued)
n = nglued*nbglued;
A = orth(rand(m,n));
A = A*diag([10.^(0:condA_glob/(n-1):condA_glob)])
      *orth(randn(n,n));
ibeg = 1;
iend = nglued;
for i=1:nbglued,
A(:,ibeg:iend) = A(:,ibeg:iend)
                  *diag([10.^(0:condA/(nglued-1):condA)])...
                  *orth(randn(nglued,nglued));
ibeg = ibeg+nglued;
iend = iend+nglued;
end
```

Here m represents the number of rows of A, *nglued* is the number of columns in a block, *nbglued* is the number of blocks that are glued together, and $n = nglued \times nbglued$ is the number of columns in the matrix. The parameter *condA* is the condition number of a block, and *condA_glob* is a parameter to couple the blocks together. The MATLAB command `orth(X)` produces an orthonormal basis for the range of X, thus the command `orth(randn(m,n))` produces a random orthogonal matrix.

For this example, we used the parameters

$$condA\_glob = 1; condA = 2; m = 200; nglued = 5; nbglued = 40;$$

for which we obtained a 200×200 matrix with condition number 506.92 (the condition number of the orthogonal factor R is about the same). We also used

Table 2 Orthogonality and Normal Equations Error from CGS Algorithms for Example 2

Algorithm	$\|A^{\mathrm{T}}A - R^{\mathrm{T}}R\|_2/\|A\|_2^2$	$\|I - Q^{\mathrm{T}}Q\|_2$
CGS–S (Algorithm 1)	3.8744e–6	9.3676e–4
CGS–P (Algorithm 2)	2.8729e–16	1.8972e–12

the command `randn('state',0)` to reset the random number generator to its initial state. Table 2 summarizes the results from applying CGS–S and CGS–P to this matrix.

For this example, the loss of orthogonality of CGS–S is far in excess of $O(\epsilon\kappa_2(R)^2)$, whereas the loss of orthogonality for CGS–P is well within that bound. The error $\|A^{\mathrm{T}}A - R^{\mathrm{T}}R\|_2$ is far larger for CGS–S than it is for CGS–P and is much greater than $O(\varepsilon_M\|A\|_2^2)$.

Remark 2 As pointed out by one referee and observed in some of our tests, when A does not satisfy the assumptions of Theorem 1 it is possible to have $\phi_k \geq \psi_k$ and thus have Algorithm 2 "break down", resulting in Q and R that are not real matrices. For Algorithm 1, such break down is not possible.

Implications for reorthogonalized Gram–Schmidt

The most common application of CGS is with reorthogonalization. The main implication of Theorem 1 is in choosing between two reorthogonalization schemes, CGS2 given in [7] and CGS-K given in [6,11]. The algorithm CGS2 [7] performs a Q–R factorization by replacing the main loop of Algorithm 1 with

$\mathbf{s}_k = Q_{k-1}^{\mathrm{T}}\mathbf{a}_k;\ \mathbf{v}_k = \mathbf{a}_k - Q_{k-1}\mathbf{s}_k;$
$\delta\mathbf{s}_k = Q_{k-1}^{\mathrm{T}}\mathbf{v}_k;\ \mathbf{v}_k = \mathbf{v}_k - Q_{k-1}\delta\mathbf{s}_k;$
$r_{kk} = \|\mathbf{v}_k\|_2;\ \mathbf{s}_k = \mathbf{s}_k + \delta\mathbf{s}_k;$
$\mathbf{q}_k = \mathbf{v}_k/r_{kk};$

Thus, CGS2 performs two orthogonalizations per step. However, two orthogonalizations are not always necessary. Instead, using a variant of procedures in for instance [1] or [4], we may choose to use the CGS-P variant for the first orthogonalization as follows

$\mathbf{s}_k = Q_{k-1}^{\mathrm{T}}\mathbf{a}_k;\ \mathbf{v}_k = \mathbf{a}_k - Q_{k-1}\mathbf{s}_k;$
$\psi_k = \|\mathbf{a}_k\|_2; \phi_k = \|\mathbf{s}_k\|_2;$
if $\|\mathbf{v}_k\|_2/\psi_k \geq \beta_{min}$
  $r_{kk} = (\psi_k - \phi_k)^{1/2}\,(\psi_k + \phi_k)^{1/2};$
else
  $\delta\mathbf{s}_k = Q_{k-1}^{\mathrm{T}}\mathbf{v}_k;\ \mathbf{v}_k = \mathbf{v}_k - Q_{k-1}\delta\mathbf{s}_k;$
  $r_{kk} = \|\mathbf{v}_k\|_2;\ \mathbf{s}_k = \mathbf{s}_k + \delta\mathbf{s}_k$
end;
$\mathbf{q}_k = \mathbf{v}_k/r_{kk};$

If we substitute this step for the main loop of Algorithm 1 or 2, this is the Q–R factorization called CGS-K by Langou [11] (see also [6] and the references therein). The value $\beta_{\min} \in (0, 1)$ is bounded away from zero (in [1], $\beta_{\min} = \sqrt{0.8}$, in [4], $\beta_{\min} = \sqrt{0.5}$). The computation of $\|\mathbf{v}_k\|_2$ in the **if** statement may be avoided by substituting the test $\phi_k/\psi_k \leq \left(1 - \beta_{\min}^2\right)^{1/2}$ thereby also preventing break down of the form discussed in Remark 2. If we take the **else** clause for each k, then we have the CGS2 algorithm. If we take the **if** clause for each k, then we have Algorithm 2. In the latter case, if A satisfies the hypothesis of Theorem 1, we have the bounds on the orthogonality of Q in (7) and on the error in the Cholesky factor R in (6). If, on the other hand, we substitute $r_{kk} = \|\mathbf{v}_k\|_2$ for the computation of r_{kk} in the **if** clause, no result like Theorem 1 is guaranteed. Therefore this common strategy for classical Gram–Schmidt works best if the normalization in Algorithm 2 is used.

Conclusion

The upper triangular factor R from classical Gram–Schmidt has been shown to satisfy the bound (5) provided that the diagonal elements of R are computed as they are in the Cholesky factorization of the normal equations matrix. If these diagonal elements are computed as in standard versions of classical Gram–Schmidt, no bounds such as (5) or (7) may be guaranteed. When developing algorithms for classical Gram–Schmidt similar to those in [1,4] with reorthogonalization, this result shows that a useful Q–R factorization is obtained even if such algorithms always stop after one orthogonalization step.

Appendix: Proof of Theorem 1

To set up the proof of Theorem 1, we require a lemma.

Lemma 2 *Let $Q \in \mathbb{R}^{m\times n}$ and $R \in \mathbb{R}^{n\times n}$ be the results of Algorithm 2 in floating point arithmetic with machine unit ε_M and that R satisfies (3). Then*

$$r_{11} = \|\mathbf{a}_1\|_2(1+\delta_1), \quad |\delta_1| \leq (0.5m+1)\varepsilon_M + O(\varepsilon_M^2) \tag{11}$$

and for $k = 2, \ldots, n$

$$r_{kk} = \left(\|\mathbf{a}_k\|_2^2(1+\delta_k) - \|\mathbf{s}_k\|_2^2(1+\Delta_k)\right)^{1/2}, \tag{12}$$

$$|\delta_k|, |\Delta_k| \leq (m+8)\varepsilon_M + O(\varepsilon_M^2),$$

$$\|\mathbf{s}_k\|_2 \leq \|\mathbf{a}_k\|_2(1+\zeta), \quad |\zeta| \leq (m+2)\varepsilon_M + O(\varepsilon_M^2). \tag{13}$$

Proof Equation (11) is just the error in the computation of $\|\mathbf{a}_1\|_2$. In the computation of $r_{kk}, k = 2, \ldots, n$, note that

$$\psi_k = f\ell(\|\mathbf{a}_k\|_2) = \|\mathbf{a}_k\|_2(1 + \epsilon_1^{(k)}), \tag{14}$$

$$\phi_k = f\ell(\|\mathbf{s}_k\|_2) = \|\mathbf{s}_k\|_2(1 + \epsilon_2^{(k)}), \tag{15}$$

$$|\epsilon_i^{(k)}| \leq (0.5m + 1)\varepsilon_M + O(\varepsilon_M^2), \quad i = 1, 2.$$

Using (3), we conclude that R is nonsingular, thus $r_{kk} > 0$ for all k. Thus in Algorithm 2, $r_{kk} > 0$ only if $\psi_k > \phi_k$.

To get (12), note that

$$r_{kk} = \sqrt{\psi_k - \phi_k}\sqrt{\psi_k + \phi_k}(1 + \epsilon_3^{(k)}), \quad |\epsilon_3^{(k)}| \leq 3\varepsilon_M + O(\varepsilon_M^2).$$

Thus using (14) and (15), we have

$$\begin{aligned} r_{kk} &= \sqrt{\|\mathbf{a}_k\|_2^2(1 + \epsilon_1^{(k)})^2 - \|\mathbf{s}_k\|_2^2(1 + \epsilon_2^{(k)})^2}(1 + \epsilon_3^{(k)}) \\ &= \left(\|\mathbf{a}_k\|_2^2(1 + \delta_k) - \|\mathbf{s}_k\|_2^2(1 + \Delta_k)\right)^{1/2} \end{aligned}$$

where

$$\begin{aligned} \delta_k &= (1 + \epsilon_1^{(k)})^2(1 + \epsilon_3^{(k)})^2 - 1, \\ \Delta_k &= (1 + \epsilon_2^{(k)})^2(1 + \epsilon_3^{(k)})^2 - 1. \end{aligned}$$

That yields

$$|\delta_k|, |\Delta_k| \leq (m + 8)\varepsilon_M + O(\varepsilon_M^2).$$

Therefore r_{kk} satisfies (12).

Since $\psi_k > \phi_k$ as outlined above, from (14) to (15), we have

$$\psi_k = \|\mathbf{a}_k\|_2(1 + \epsilon_1^{(k)}) > \phi_k = \|\mathbf{s}_k\|_2(1 + \epsilon_2^{(k)})$$

thus

$$\begin{aligned} \|\mathbf{s}_k\|_2 &< \|\mathbf{a}_k\|_2(1 + \epsilon_1^{(k)})(1 + \epsilon_2^{(k)})^{-1} \\ &\leq \|\mathbf{a}_k\|_2(1 + \zeta) \end{aligned}$$

where ζ satisfies (13). □

As a consequence of the singular value version of the Cauchy interlace theorem [8, p. 449–450, Corollary 8.6.3], we have that $\|R_k\|_2 \leq \|R\|_2$ and $\|R_k^{-1}\|_2 \leq \|R^{-1}\|_2$. We will use these facts freely in the proof of Theorem 1.

We can now prove Theorem 1.

Proof [of Theorem 1] The results (4)–(5) are proven by induction on k. First, consider $k = 1$. From Lemma 2, we have (11), so

$$r_{11} = \|\mathbf{a}_1\|_2(1+\delta_1), \quad |\delta_1| \leq (0.5m+1)\varepsilon_M + O(\varepsilon_M^2)$$

which implies that

$$\begin{aligned} R_1^{\mathrm{T}} R_1 = r_{11}^2 &= \|\mathbf{a}_1\|_2^2(1+\delta_1)^2 \\ &= A_1^{\mathrm{T}} A_1 (1+\delta_1)^2 = A_1^{\mathrm{T}} A_1 + E_1 \end{aligned}$$

where

$$E_1 = 2\delta_1 A_1^{\mathrm{T}} A_1 + \delta_1^2 A_1^{\mathrm{T}} A_1.$$

Thus

$$\|E_1\|_2 = |E_1| \leq (m+2)\|\mathbf{a}_1\|_2^2 \varepsilon_M + O(\varepsilon_M^2) = (m+2)\|A_1\|_2^2 \varepsilon_M + O(\varepsilon_M^2).$$

Also, we can conclude from standard error bounds that

$$\mathbf{q}_1 = (I + G_1)\mathbf{a}_1/r_{11}, \quad \|G_1\|_2 \leq \varepsilon_M.$$

Therefore

$$A_1 - Q_1 R_1 = \mathbf{a}_1 - \mathbf{q}_1 r_{11} = -G_1 \mathbf{a}_1$$

so that

$$\|A_1 - Q_1 R_1\|_2 = \|\mathbf{a}_1 - \mathbf{q}_1 r_{11}\|_2 \leq \|G_1\|_2 \|\mathbf{a}_1\|_2 \leq \varepsilon_M \|\mathbf{a}_1\|_2. \tag{16}$$

Assume that (4)–(8) hold for $k-1$, and prove them for k. We first prove (4),(5), and then show that (6)–(8) follow.

First, we start with error bounds of the computation of the vectors $\mathbf{s}_k$,$\mathbf{v}_k$, and $\mathbf{q}_k$ to prove (4). Note that

$$\mathbf{s}_k = f\ell(Q_{k-1}^{\mathrm{T}} \mathbf{a}_k) = Q_{k-1}^{\mathrm{T}} \mathbf{a}_k - \delta \mathbf{s}_k \tag{17}$$

where

$$\begin{aligned} \|\delta \mathbf{s}_k\|_2 &\leq m\sqrt{k-1}\|Q_{k-1}\|_2 \|\mathbf{a}_k\|_2 \varepsilon_M + O(\varepsilon_M^2) \\ &\leq \sqrt{2(k-1)}m\|\mathbf{a}_k\|_2 \varepsilon_M + O(\varepsilon_M^2). \end{aligned} \tag{18}$$

Also, we have

$$\mathbf{v}_k = f\ell(\mathbf{a}_k - Q_{k-1}\mathbf{s}_k) = \mathbf{a}_k - Q_{k-1}\mathbf{s}_k - \delta\mathbf{v}_k \tag{19}$$

where

$$\|\delta\mathbf{v}_k\|_2 \leq \|\mathbf{a}_k\|_2\varepsilon_M + \sqrt{k-1}m\|Q_{k-1}\|_2\|\mathbf{s}_k\|_2\varepsilon_M + O(\varepsilon_M^2).$$

From (13), the bound on $\|\mathbf{s}_k\|_2$ in (13), and the induction hypothesis on Q_{k-1}, we have

$$\|\delta\mathbf{v}_k\|_2 \leq (\sqrt{2(k-1)}m + 1)\|\mathbf{a}_k\|_2\varepsilon_M + O(\varepsilon_M^2). \tag{20}$$

Again using the bound on $\|\mathbf{s}_k\|_2$ in (13), we note that

$$\begin{aligned}
\|\mathbf{v}_k + \delta\mathbf{v}_k\|_2{}^2 &= \|\mathbf{a}_k\|_2^2 - 2\mathbf{a}_k^{\mathrm{T}}Q_{k-1}\mathbf{s}_k + \|Q_{k-1}\mathbf{s}_k\|_2^2 \\
&= \|\mathbf{a}_k\|_2^2 - 2\|\mathbf{s}_k\|_2^2 + \|Q_{k-1}\mathbf{s}_k\|_2^2 - 2(\delta\mathbf{s}_k)^{\mathrm{T}}\mathbf{s}_k \\
&\leq \|\mathbf{a}_k\|_2^2 - 2\|\mathbf{s}_k\|_2^2 + \|Q_{k-1}\|_2^2\|\mathbf{s}_k\|_2^2 - 2(\delta\mathbf{s}_k)^{\mathrm{T}}\mathbf{s}_k \\
&\leq \|\mathbf{a}_k\|_2^2 - 2\|\mathbf{s}_k\|_2^2 + 2\|\mathbf{s}_k\|_2^2 - 2(\delta\mathbf{s}_k)^{\mathrm{T}}\mathbf{s}_k \\
&= \|\mathbf{a}_k\|_2^2 - 2(\delta\mathbf{s}_k)^{\mathrm{T}}\mathbf{s}_k \\
&\leq \|\mathbf{a}_k\|_2^2 + 2\|\delta\mathbf{s}_k\|_2\|\mathbf{s}_k\|_2 \\
&= \|\mathbf{a}_k\|_2^2 + 2\|\delta\mathbf{s}_k\|_2\|\mathbf{a}_k\|_2 + O(\varepsilon_M^2) \\
&\leq \|\mathbf{a}_k\|_2^2(1 + \sqrt{2(k-1)}m\varepsilon_M)^2 + O(\varepsilon_M^2).
\end{aligned}$$

Thus

$$\|\mathbf{v}_k\|_2 \leq \|\mathbf{a}_k\|_2(1 + (3\sqrt{2(k-1)}m)\varepsilon_M) + O(\varepsilon_M^2) = \|\mathbf{a}_k\|_2 + O(\varepsilon_M).$$

We note that

$$\mathbf{q}_k = (I + G_k)\mathbf{v}_k/r_{kk}, \quad \|G_k\|_2 \leq \varepsilon_M.$$

If we let

$$\Delta A_k = Q_kR_k - A_k$$

then

$$\Delta A_k = \left(\Delta A_{k-1}\ \delta\mathbf{a}_k\right)$$

where

$$\begin{aligned}
\delta\mathbf{a}_k &= (I + G_k)\mathbf{v}_k + Q_{k-1}\mathbf{s}_k - \mathbf{a}_k, \\
&= G_k\mathbf{v}_k - \delta\mathbf{v}_k.
\end{aligned}$$

That yields

$$\|\delta \mathbf{a}_k\|_2 \le \|G_k\|_2 \|\mathbf{v}_k\|_2 + \|\delta \mathbf{v}_k\|_2 \le (2\sqrt{2(k-1)}m + 2)\varepsilon_M \|\mathbf{a}_k\|_2 + O(\varepsilon_M^2).$$

To bound $\|\Delta A_k\|_2$, we give a recurrence for bounding $\|\Delta A_k\|_F$ in terms of $\|A_k\|_F$, then use the bound $\|A_k\|_F \le \sqrt{k}\|A_k\|_2$. We show that

$$\|\Delta A_k\|_F \le \hat{c}_1(m,k)\|A_k\|_F \varepsilon_M + O(\varepsilon_M^2).$$

For $k = 1$,

$$\|\Delta A_1\|_F = \|\mathbf{a}_1\|_2 = \varepsilon_M \|\mathbf{a}_1\|_2 = \varepsilon_M \|A_1\|_F.$$

Using properties of the Frobenius norm,

$$\begin{aligned}
\|\Delta A_k\|_F^2 &\le \|\Delta A_{k-1}\|_F^2 + \|\delta \mathbf{a}_k\|_2^2 \\
&\le [\hat{c}_1^2(m,k-1)\|A_{k-1}\|_F^2 + (2\sqrt{2(k-1)}m+2)^2 \|\mathbf{a}_k\|_2^2]\varepsilon_M^2 + O(\varepsilon_M^3) \\
&\le \max\{\hat{c}_1^2(m,k-1), (2\sqrt{2(k-1)}m+2)^2\}(\|A_{k-1}\|_F^2 + \|\mathbf{a}_k\|_2^2)\varepsilon_M^2 \\
&\quad + O(\varepsilon_M^3) \\
&= \hat{c}_1^2(m,k)\|A_k\|_F^2 \varepsilon_M^2 + O(\varepsilon_M^3).
\end{aligned} \tag{21}$$

A quick induction argument yields

$$\hat{c}_1(m,k) = 2\sqrt{2(k-1)}m + 2 \le 2\sqrt{2k}m + 2.$$

Thus

$$\|\Delta A_k\|_2 \le \|\Delta A_k\|_F \le \hat{c}_1(m,k)\varepsilon_M \|A_k\|_F + O(\varepsilon_M^2) \le \sqrt{k}\hat{c}_1(m,k)\|A_k\|_2 + O(\varepsilon_M^2)$$

yielding (4) with $c_1(m,k) = 2\sqrt{2}mk + 2\sqrt{k} \ge \sqrt{k}\hat{c}_1(m,k)$.

To prove (5), note that

$$E_k = R_k^{\mathrm{T}} R_k - A_k^{\mathrm{T}} A_k = \begin{array}{c c} & \begin{array}{cc} k-1 & 1 \end{array} \\ \begin{array}{c} k-1 \\ 1 \end{array} & \begin{pmatrix} E_{k-1} & \mathbf{w}_k \\ \mathbf{w}_k^T & e_{kk} \end{pmatrix} \end{array}$$

where using Lemma 2, we have

$$\begin{aligned}
\mathbf{w}_k &= R_{k-1}^{\mathrm{T}} \mathbf{s}_k - A_{k-1}^{\mathrm{T}} \mathbf{a}_k, \\
e_{kk} &= \mathbf{s}_k^{\mathrm{T}} \mathbf{s}_k + r_{kk}^2 - \mathbf{a}_k^{\mathrm{T}} \mathbf{a}_k \\
&= \delta_k \mathbf{a}_k^{\mathrm{T}} \mathbf{a}_k - \Delta_k \mathbf{s}_k^{\mathrm{T}} \mathbf{s}_k.
\end{aligned}$$

Using the bounds on δ_k and Δ_k in (12), we have

$$\begin{aligned}|e_{kk}| &\le |\delta_k|\|\mathbf{a}_k\|_2^2 + |\Delta_k|\|\mathbf{s}_k\|_2^2\\ &\le (|\delta_k| + |\Delta_k|)\|\mathbf{a}_k\|_2^2 + O(\varepsilon_M^2)\\ &\le 2(m+8)\|\mathbf{a}_k\|_2^2\varepsilon_M + O(\varepsilon_M^2)\\ &\le 2(m+8)\|A_k\|_2^2\varepsilon_M + O(\varepsilon_M^2).\end{aligned}$$

Since

$$\mathbf{s}_k + \delta\mathbf{s}_k = Q_{k-1}^{\mathrm{T}}\mathbf{a}_k, \quad A_{k-1} + \Delta A_{k-1} = Q_{k-1}R_{k-1}$$

we have

$$\begin{aligned}\mathbf{w}_k &= R_{k-1}^{\mathrm{T}}\mathbf{s}_k - A_{k-1}^{\mathrm{T}}\mathbf{a}_k\\ &= R_{k-1}^{\mathrm{T}}Q_{k-1}^{\mathrm{T}}\mathbf{a}_k - R_{k-1}^{\mathrm{T}}\delta\mathbf{s}_k - A_{k-1}^{\mathrm{T}}\mathbf{a}_k\\ &= \Delta A_{k-1}^{\mathrm{T}}\mathbf{a}_k - R_{k-1}^{\mathrm{T}}\delta\mathbf{s}_k. \end{aligned} \tag{22}$$

So that $\|\mathbf{w}_k\|_2$ has the bound

$$\begin{aligned}\|\mathbf{w}_k\|_2 &\le \|\Delta A_{k-1}\|_2\|\mathbf{a}_k\|_2 + \|R_{k-1}\|_2\|\delta\mathbf{s}_k\|_2 + O(\varepsilon_M^2)\\ &\le (c_1(m,k-1)\|A_{k-1}\|_2\|\mathbf{a}_k\|_2 + \sqrt{2(k-1)}m\|A_{k-1}\|_2\|\mathbf{a}_k\|_2)\varepsilon_M\\ &\le [2\sqrt{2}m(k-1) + 2\sqrt{k-1} + \sqrt{2(k-1)}m]\|A_{k-1}\|_2\|\mathbf{a}_k\|_2\varepsilon_M + O(\varepsilon_M^2)\\ &\le 7m(k-1)\|A_k\|_2^2\varepsilon_M + O(\varepsilon_M^2)\end{aligned} \tag{23}$$

We have that

$$\begin{aligned}\|E_k\|_2 &\le \left\| \begin{pmatrix} E_{k-1} & 0\\ 0 & e_{kk}\end{pmatrix} \right\|_2 + \left\| \begin{pmatrix} 0 & \mathbf{w}_k\\ \mathbf{w}_k^T & 0\end{pmatrix} \right\|_2\\ &\le \max\{\|E_{k-1}\|_2, |e_{kk}|\} + \|\mathbf{w}_k\|_2\\ &\le [\max\{c_2(m,k-1), 2(m+8)\} + 7m(k-1)]\|A_k\|_2^2\varepsilon_M + O(\varepsilon_M^2)\\ &< [c_2(m,k-1) + 2(m+8) + 7m(k-1)]\|A_k\|_2^2\varepsilon_M + O(\varepsilon_M^2)\\ &\le c_2(m,k)\|A_k\|_2^2\varepsilon_M + O(\varepsilon_M^2)\end{aligned} \tag{24}$$

where

$$\begin{aligned}c_2(m,k) &= \sum_{j=1}^{k}[2(m+8) + 7m(j-1)]\\ &= 3.5m(k-1)k + 2mk + 16k.\end{aligned}$$

Thus we have the expression for $c_2(m,k)$ given in Eq. (2).

To prove (6)–(8), we simply apply (4), (5). Equation (6) results from noting that

$$\begin{aligned}\|R_k\|_2^2 &= \|R_k^{\mathrm{T}} R_k\|_2 = \|A_k^{\mathrm{T}} A_k + E_k\|_2 \\ &\le \|A_k^{\mathrm{T}} A_k\|_2 + \|E_k\|_2 \le (1 + c_2(m,k)\varepsilon_M)\|A_k\|_2^2 + O(\varepsilon_M^2).\end{aligned}$$

Thus,

$$\|R_k\|_2 \le (1 + c_3(m,k)\varepsilon_M)\|A_k\|_2 + O(\varepsilon_M^2)$$

where

$$1 + c_3(m,k)\varepsilon_M + O(\varepsilon_M^2) = \sqrt{1 + c_2(m,k)},$$

that is, $c_3(m,k) = 0.5c_2(m,k)$. Reversing the roles of R_k and A_k yields

$$\|A_k\|_2 \le (1 + c_3(m,k)\varepsilon_M)\|R_k\|_2 + O(\varepsilon_M^2),$$

thus we have (6).

To get (7), we note that

$$Q_k = (A_k + \Delta A_k)R_k^{-1}$$

so that

$$\begin{aligned}I - Q_k^{\mathrm{T}} Q_k &= R_k^{-\mathrm{T}}(R_k^{\mathrm{T}} R_k - (A_k + \Delta A_k)^{\mathrm{T}}(A_k + \Delta A_k))R_k^{-1} \\ &= R_k^{-\mathrm{T}}(E_k - A_k^{\mathrm{T}} \Delta A_k - (\Delta A_k)^{\mathrm{T}} A_k - (\Delta A_k)^{\mathrm{T}}(\Delta A_k))R_k^{-1}.\end{aligned}$$

Thus

$$\begin{aligned}\|I - Q_k^{\mathrm{T}} Q_k\|_2 &\le \|R_k^{-1}\|_2^2(\|E_k\|_2 + 2\|\Delta A_k\|_2\|A_k\|_2 + \|\Delta A_k\|_2^2) \\ &\le \|R_k^{-1}\|_2^2(c_2(m,k)\|A_k\|_2^2 \\ &\quad + 2c_1(m,k)\|A_k\|_2^2 + \varepsilon_M c_1^2(m,k)\|A_k\|_2^2)\varepsilon_M + O(\varepsilon_M^2) \\ &\le \|R_k\|_2^2\|R_k^{-1}\|_2^2(c_2(m,k) + 2c_1(m,k))\varepsilon_M + O(\varepsilon_M^2) \\ &= c_4(m,k)\|R_k\|_2^2\|R_k^{-1}\|_2^2\varepsilon_M + O(\varepsilon_M^2)\end{aligned}$$

where $c_4(m,k) = c_2(m,k) + 2c_1(m,k)$.

Finally, to get (8), we have that

$$\begin{aligned}\|Q_k\|_2^2 &= \|Q_k^{\mathrm{T}} Q_k\|_2 = \|I - Q_k^{\mathrm{T}} Q_k - I\|_2 \\ &\le \|I\|_2 + \|I - Q_k^{\mathrm{T}} Q_k\|_2 \\ &\le 1 + \|I - Q_k^{\mathrm{T}} Q_k\|_2 \\ &\le 1 + c_4(m,k)\|R_k\|_2^2 \|R_k^{-1}\|_2^2 \varepsilon_M + O(\varepsilon_M^2) \le 2 + O(\varepsilon_M^2).\end{aligned}$$

Taking square roots yields (8). □

References

1. Barlow, J.L., Smoktunowicz, A., Erbay, H.: Improved Gram–Schmidt downdating methods. BIT **45**, 259–285 (2005)
2. Björck, Å.: Numerical methods for least squares problems. SIAM, Philadelphia, (1996)
3. Björck, Å.: Solving linear least squares problems by Gram–Schmidt orthogonalization. BIT, **7**, 1–21 (1967)
4. Daniel, J.W., Gragg, W.B., Kaufman, L., Stewart, G.W.: Reorthogonalization and stable algorithms for updating the Gram–Schmidt QR factorization. Math. Comp. **30**(136), 772–795 (1976)
5. Gander, W.: Algorithms for the qr–decomposition. Technical Report 80–02, Angewandte Mathematik, ETH, 1980
6. Giraud, L., Langou, J.: Robust selective Gram-Schmidt reorthogonalization. SIAM J. Sci Comput. **25**(2), 417–441 (2003)
7. Giraud, L., Langou, J., Rozložnik, M., Van Den Eshof, J.: Rounding error analysis of the classical Gram–Schmidt orthogonalization process. Numer Math, **101**(1), 87–100 (2005)
8. Golub, G.H., Van Loan, C.F.: Matrix Computations, Third edn. The Johns Hopkins Press, Baltimore (1996)
9. Hoffmann, W.: Iterative algorithms for Gram–Schmidt orthogonalization. Computing **41**, 353–367 (1989)
10. Kiełbasiński, A.: Analiza numeryczna algorytmu ortogonalizacji Grama–Schmidta (in Polish). Roczniki Polskiego Towarzystwa Matematycznego, Seria III: Matematyka Stosowana, **II**, 15–35 (1974)
11. Langou, J.: CGS-P: A new CGS Algorithm in $\epsilon \cdot \kappa(A)^2$ Unpublished Power Point Presentation, Ninth Copper Mountain Conference on Iterative Methods,Copper Mountain, April 2006
12. Overton, M.L.: Numerical Computing with IEEE Floating Point Arithmetic. SIAM Publications, Philadelphi, (2001)
13. Parlett, B.N.: The Symmetric Eigenvalue Problem. SIAM, Philadelphia, (1998) Republication of 1980 book
14. Wolcendorf, M.: Modifying the Q–R decomposition. Master's thesis, Warsaw University of Technology, 2001 (in Polish)